a No/Fee

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: CHRISTOPHER J. EDGE ET AL.
Docket: 4362.37US02
Title: ARRANGEMENT FOR MAPPING COLORS BETWEEN IMAGING SYSTEMS AND METHOD THEREFOR

CERTIFICATE UNDER 37 CFR 1.10
'Express Mail' mailing label number: EM296798633US
Date of Deposit: JULY 3, 1997
I hereby certify that this paper or fee is being deposited with the United States Postal Service 'Express Mail Post Office To Addressee' service under 37 CFR 1.10 and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

By:____________________
Name: Nassir Mohamoud

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

We are transmitting herewith the attached:

☒ Transmittal sheet, in duplicate, containing Certificate under 37 CFR 1.10.
☒ Utility Patent Application: Spec. 37 pgs; 24 claims; Abstract 1 pg.:
☒ Six sheets of informal drawings
☒ An unsigned Combined Declaration and Power of Attorney
☒ Other: Appendix A
☒ Return postcard
☒ **PAYMENT OF THE FILING FEE IS BEING DEFERRED.**

MERCHANT, GOULD, SMITH, EDELL,
WELTER & SCHMIDT
3100 Norwest Center, Minneapolis, MN 55402
(612) 332-5300

By: RJ Crawford
Name: Robert J. Crawford
Reg. No.: 32,122
Initials: RJC:PST:bkh

(PTO TRANSMITTAL - NEW FILING/DEFERRED FEES)

M&G 4362.37US02

ARRANGEMENT FOR MAPPING COLORS BETWEEN IMAGING SYSTEMS AND METHOD THEREFOR

Field of the Invention

The present invention relates to color imaging. More particularly, the present invention relates to mapping colors between color imaging systems.

Background of the Invention

Color reproduction processes typically involve using color imaging systems to produce colors on various media. These color imaging systems may be used to duplicate a color image from one medium to another medium, *e.g.*, from one printed copy to another or from a display screen to a printed copy. Color reproduction processes are used in various application environments, for example, color proofing applications.

Some color reproduction processes use approaches known as color management systems (CMSs) to characterize various color imaging systems and to transform color data between the color imaging systems. Characterizing color imaging systems typically involves calculating color response functions using color coordinate systems known as color spaces. One commonly-used color space is Commission Internationale de l'Éclairage $L^*a^*b^*$ (CIELAB) space. CMSs attempt to reproduce an original color image on a color imaging system so as to preserve the appearance of colors between the original and the reproduction within the limitations of the color imaging system of the reproduction process.

Various CMS approaches have been proposed to achieve accurate color reproduction. Many of these approaches involve producing color samples using an output or display device and measuring the color values of the samples using an input device. Such approaches correlate the output colors with the measured color values. This correlation is performed using, for example, forward and reverse transforms between device-specific color spaces and a device-independent color space. These transformation techniques are often supplemented by interpolation between entries in a multidimensional lookup table. These techniques exhibit inaccurate color conversion between similar devices, potentially resulting in undesirable contamination of colors. Furthermore, accurate color conversion of dark colors has often been particularly difficult because of inadequate processing of black channel data in many applications.

CMSs often perform gamut mapping to correlate the range or gamut of colors that can be realized by a device with regions of a color space. Because many devices are incapable of realizing the complete range of colors in a color space, gamut mapping typically involves compressing or scaling regions of the color space. The device can then approximate colors outside its gamut using the compressed regions of the color space. For many CMSs, gamut mapping is potentially inconsistent under certain circumstances, such as when using profiles generated by software from different vendors. In addition, many CMSs exhibit inconsistencies when performing forward and reverse transformations between imaging systems. For example, color shifting often occurs with repeated forward and reverse transformations.

Many CMS techniques exhibit other limitations in addition to the lack of accuracy in converting colors. For example, many CMS techniques are relatively inflexible with respect to changes in illumination and observer conditions, gamut mapping, and choice of color space. Certain techniques lack forward compatibility with future color standards.

Summary of the Invention

According to one embodiment, the present invention is directed to a color mapping method for use in transforming colors between color imaging systems. The color mapping method includes using forward transformation profiles characterizing the color imaging systems to generate respective sets of device-independent color values for the color imaging systems. Color conversions are calculated by recursively reducing differences between the sets of device-independent color values. This difference reduction is also optionally performed on black channel information to obtain a mapping of black channels between the color imaging systems. A color map describing a relationship between the color imaging systems is constructed as using the predicted color conversions. This method may be performed by a color mapping arrangement or a computer-executable program stored on a data storage medium.

According to another embodiment of the present invention, color mapping between source and destination color imaging systems is accomplished by using profiles that characterize the color imaging systems to generate device-independent color values for the source color imaging system and to convert to device-dependent values of the destination color imaging system by performing a color conversion using the profiles. The device-independent color values have a same dimensionality as the corresponding color imaging systems. The color conversion can be used to improve its own accuracy relative to a quality threshold. The color conversion is used to define a color map for transforming colors between the color imaging systems.

Another embodiment of the present invention is directed to a color mapping arrangement for use in transforming colors between imaging systems. A computer arrangement uses forward transformation profiles that characterize the color imaging systems to generate respective sets of device-independent color values for the color imaging systems. The computer arrangement also calculates color conversions by recursively reducing differences between the sets of device-independent color values. The computer arrangement uses the color conversions to construct a color map describing a relationship between the color imaging systems using the color conversions. A memory stores the color map.

The above summary of the invention is not intended to describe each disclosed embodiment of the present invention. This is the purpose of the figures and of the detailed description that follows.

Brief Description of the Drawings

Other aspects and advantages of the present invention will become apparent upon reading the following detailed description and upon reference to the drawings in which:

FIG. 1 is a block diagram illustrating an example color mapping system, according to an embodiment of the present invention;

FIG. 2 is a block diagram illustrating an example arrangement implementing part of the color mapping system of FIG. 1, according to an embodiment of the present invention;

FIG. 3 is a block diagram illustrating another example arrangement implementing part of the color mapping system of FIG. 1, according to an embodiment of the present invention;

FIG. 4 is a block diagram illustrating yet another example arrangement implementing part of the color mapping system of FIG. 1, according to an embodiment of the present invention;

FIG. 5 is a block diagram illustrating still another example arrangement implementing part of the color mapping system of FIG. 1, according to an embodiment of the present invention; and

FIG. 6 is a flow chart illustrating an example color mapping method, according to another embodiment of the present invention.

While the invention is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It should be understood, however, that the intention is not to limit the invention to the particular embodiments described. On the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

Detailed Description of the Various Embodiments

The present invention is believed to be applicable to a variety of systems and arrangements that characterize color imaging systems. The invention has been found to be particularly advantageous for transforming colors between different color imaging systems. An appreciation of various aspects of the invention is best gained through a discussion of these particular application examples.

According to one aspect of the present invention, a color mapping technique may be applied to a variety of color imaging systems to generate a color map that can be used to transform the color response of one color imaging system, referred to as a source color imaging system, to match the color response of another color imaging system, referred to as a destination color imaging system. The color mapping technique projects color coordinates in the color space used by the source color imaging system into, for example, a device-independent color space. Optimal color coordinates in the color space used by the destination color imaging system are determined that realize a relatively close match between the projections into the device-independent color space of the color coordinates in the color space used by the source color imaging system and the optimal color coordinates. The color mapping technique then generates a color map based on the optimal color coordinates for a number of color coordinates in the color space used by the source color imaging system.

FIG. 1 illustrates an example system 100 according to the present invention configured to transform colors between imaging systems. The system 100 includes an appropriately-programmed computer arrangement 102. The computer arrangement 102 may be implemented using any of a variety of conventional resources, for example, a personal computer and CD-ROM based software. Other computer-based designs may be used as well. For example, the computer arrangement 102 may be implemented using a microprocessor that acts as a read-only memory (ROM) into which a software application program is loaded. The software application program may be incorporated, for example, in a color-management software package.

A color mapping system 104 includes a color management system 106. The color management system 106 receives a source device profile 108 and a destination device profile 110. These device profiles describe mappings from device-dependent color coordinate systems used by respective color imaging systems to device-independent color coordinate systems.

The color management system 106 processes the source device profile 108 and the destination device profile 110 to generate a color map 114. The color map 114 describes a relationship between the color imaging systems used by the source and destination devices. A memory 116 stores the color map 114. Subsequently, the color management system 106 uses the color map 114 to transform a set of source coordinates 118 in a device-dependent source device color space into a set of destination coordinates 120 in a device-dependent destination device color space.

FIG. 2 illustrates an example color management system 200 for transforming colors between imaging systems according to the present invention. A source device profile interpreter 202 receives a source device profile 206. The source device profile 206 is used to map coordinates in the source device color space to a some form of color data, such as spectral or XYZ tristimulus values. For example, if the source device is a halftone color printer, the source device profile 206 may map CMYK color values to a XYZ color space. The source device profile interpreter 202 interprets the source device profile 206 and converts coordinates in the source device color space to a device-independent color space known as a profile connecting space (PCS). The PCS is used for converting the coordinates in the source device color space to the destination device color space. The PCS may be, for example, the CIELAB color space. Another example PCS is described in copending U.S. Patent Application, entitled "Characterization of Color Imaging Systems" (Christopher Edge et al.), assigned to the instant assignee, filed on June 27, 1997, and incorporated herein by reference.

A destination device profile interpreter 208 receives a destination device profile 210. The destination device profile 210 is used to map color coordinates in a destination device color space used by a destination device 212 to some form of color data, such as spectral or XYZ tristimulus values. For example, if the destination device 212 is a cathode ray tube (CRT) monitor, the destination device profile 210 may map color coordinates in a red-green-blue (RGB) color space to XYZ tristimulus values. The destination device profile interpreter 208 interprets the destination device profile 210 and converts color coordinates in the destination device color space to the PCS.

The source and destination device profile interpreters 202 and 208 may be implemented using any of a variety of hardware and software arrangements and are configurable for a variety of application environments. For example, if the source and destination device profiles 206 and 210 are International Color Consortium (ICC) device profiles, the source and destination device profile interpreters 202 and 208 are optionally configured to include white- and black-point parameters to account for color variations between media and colorants used by different color display devices. The source and destination device profile interpreters 202 and 208 can also be configured to include pleasing color corrections, such as L^* rescaling and a^*b^* hue adjustments. Alternatively, the pleasing color corrections can be incorporated into the color transformer 214. In certain other application environments, the source and destination device profile interpreters 202 and 208 are further configurable to include, for example, illuminant and observer functions.

The device profile interpreters 202 and 208 can be configured using any of a variety of approaches. For example, plug-in software modules can be used to configure the device profile interpreters 202 and 208. Using plug-in software modules obviates the need to use new versions of the color management system 200 or of the device profiles 206 and 210 when adding, for example, a newly defined color space, a custom illuminant, such as fluorescent light, or a new gamut mapping technique. These options can be selected, *e.g.*, using a setup window at the operating system level. For example, if the operating system is Apple OS version 7.5, these options can be selected using a control panel interface.

If the device characterization is non-spectral, the color management system 200 can use the original spectral data that is saved with the profile to reconstruct the device profiles according to various conditions, such as illuminant functions and color space choices. For example, if one uses an RGB regression to convert scanner RGB values into color space values for a particular combination of color space and illuminant and observer conditions based on a set of spectral data, the regression for a new set of conditions can be generated based on the same spectral data. Accordingly, the device profiles 206 and 210 can be used to calculate color values for a variety of conditions and color appearance models.

A color transformer 214 obtains PCS color coordinates from the source and destination device profile interpreters 202 and 208. The color transformer 214 uses these color coordinates to develop a color map 216 that expresses a relationship between the color spaces used by the source and destination devices 204 and 212. To generate the color map 216, the color transformer 214 may use any of a variety of gamut mapping techniques. One such technique that has been found to yield particularly accurate results involves reducing the color error between the source and destination devices. The color error is defined, for example, by Euclidean distances in the PCS or by weighted sum square errors in a color space that is polar in the chromatic dimensions of the PCS. Defining the color error using weighted sum square errors results in a mapping between color imaging systems that accurately maintains colors in reproduced images. By using error reduction techniques, the color transformer 214 avoids generating significant cumulative error in performing multiple forward and reverse transformations between color spaces.

The color transformer 214 is implemented using, for example, a software program, and can be configured for a variety of applications. For example, the color transformer 214 can be configured to perform a 100% black point scaling for mapping a printed color image to a monitor display of the image. On the other hand, because newsprint has a relatively weak black point attributable to its ink density and light-transmitting properties, the color transformer 214 can be configured to perform, for example, a 50% black point scaling when mapping a color image printed on newsprint to the Matchprint™ color imaging system. The color transformer 214 is also configurable to use, for example, illuminant and observer functions, which the color transformer 214 provides to the source and destination device profile interpreters 202 and 208. The color management system 200 receives user preferences from an input 218 to determine how to configure the color transformer 214.

After developing the color map 216, the color transformer 214 can be used to transform colors between the source and destination devices 204 and 212. The color transformer 214 receives color coordinates from the source device 204 and transforms them using the color map 216. This transformation produces a set of color coordinates in the destination device color space. The destination color imaging system then reproduces the color on the destination device 212 using these color coordinates.

FIG. 3 illustrates an example device profile interpreter 300 implementing part of the color management system 200 of FIG. 2. The device profile interpreter 300 uses a device profile 302 to convert device coordinates received at an input 304 to PCS color coordinates, which the device profile interpreter 300 provides at an output 306. The device profile 302 describes the relationship between the device coordinates and some form of color data. Additionally, the device profile 302 optionally stores the raw spectral data used to construct the device profile 302. The raw spectral data allows subsequent construction of more accurate device profiles 302, *e.g.*, if ICC specifications change. This updating can be performed automatically, for example, upon detecting that some component of the device profile 302 is out of date. Updates can also be performed periodically based on a schedule. To update the device profile 302, a new profile can be generated using the spectral data. Alternatively, error reduction, such as a one-dimensional correction, can be performed on each channel in the original look-up table for constructing the new profile. This correction can be applied as a separate set of one-dimensional tables or applied directly to the analytical model or multidimensional look-up table. For additional information concerning an example error reduction procedure that can be used in constructing a new profile, reference can be made to U.S. Patent Application Ser. No. 08/431,614, entitled "Apparatus and Method for Recalibrating a Multi-Color Imaging System," assigned to the instant assignee and incorporated herein by reference.

A device profile processor 308 receives the device coordinates from the input 304 and the device profile 302. The device profile 302 may be, for example, an ICC profile. If the device profile 302 exists in this format, the device profile processor receives the forward portion of the profile, *i.e.*, the portion used for converting device coordinates to PCS color coordinates. Alternatively, the device profile 302 can be stored in another format. The device profile processor 308 processes the device coordinates using the device profile 302 and outputs certain data based on the device profile 302. For example, if the device profile 302 is an ICC profile, the device profile processor 308 outputs XYZ tristimulus values for a particular set of observer conditions (*e.g.*, illuminant and observer functions). If the device profile 302 is based on spectral data, the device profile processor 308 outputs spectral data. The device profile processor 308 can be configured for a variety of applications. For example, a user can select between absolute and relative colorimetrics and can configure observer, *e.g.*, illuminant, conditions.

A PCS processor 310 receives the data output from the device profile processor 308 and a set of PCS parameters from an input 312. The PCS parameters may include, for example, XYZ tristimulus values for the media white, the illuminant white, and the black point, as well as black-point scaling from a perfect black to the media black. The PCS processor 310 generates the PCS values as a function of the data received from the device profile processor 308 and the PCS parameters.

FIG. 4 illustrates an example color transformer 400 implementing part of the color management system 200 of FIG. 2. A device link generator 402 receives as input at least one source profile and one destination profile. While FIG. 4 illustrates the device link generator 402 receiving a source device profile from a source profile interpreter 404 and a destination device profile from a destination profile interpreter 406, it should be understood that the device link generator 402 may also receive one or more device profiles that are intermediate between the source and destination device profiles. For example, a device profile characterizing an RGB monitor can be intermediate between a source device profile characterizing an RGB scanner and a destination device profile characterizing a CMYK printer. The source and destination device profiles are forward transforms and optionally include configurable observer conditions and PCS parameters. The device link generator 402 also receives a series of PCS parameters 408 to improve linking of different device types (*e.g.*, CRT monitors and printers). The gamut mapping parameters 410 improve mapping of out of gamut colors between device types.

The device link generator 402 generates a color map or device profile link 412 that maps colors between two devices, *e.g.*, from an RGB device to a CMYK device or between two CMYK devices. The device profile link 412 is, for example, a mathematical expression or a look-up table. The color transformer 400 optionally stores the device profile link 412 in a memory, such as a random access memory (RAM), or saves it as a file for multiple transformations between the source and destination device color spaces.

A device link calculator 414 receives source device coordinates from an input 416 and processes them using the device profile link 412. The device link calculator 414 uses a single forward calculation to transform the source device coordinates to a set of destination device coordinates for presentation at an output 418. Because the device link calculator 414 uses a single forward calculation, interpolation is relatively simple and easily optimized and the transformation process is relatively fast. If the device profile link 412 is a look-up table, the device link calculator 414 optionally uses linear interpolation to refine the destination device coordinates. The device link calculator 414 can be implemented, *e.g.*, using a conventional multidimensional linear interpolator.

FIG. 5 illustrates an example device link generator 500 that implements part of the color transformer 400 of FIG. 4. The device link generator 500 includes a device link table builder 502 that creates a look-up table to enable rapid interpolation of destination device coordinates from source device coordinates. It should be understood that if the device profile link is a mathematical expression rather than as a look-up table, an analogous transformation generator replaces the device link table builder 502. Such a transformation builder may, for example, generate coefficients for use in the mathematical expression. To facilitate the discussion, however, the device link generator 500 is assumed to include a device link table builder 502. The device link table builder 502 generates the look-up table by generating a series of source device coordinates as input value entries and determining the optimal destination device coordinates as output values corresponding to the input values. The device link table builder 502 generates all combinations of source device coordinates using, for example, a series of nested loops, one loop for each dimension of the source device color coordinate space.

To reduce the computational and memory requirements for constructing and storing the look-up table, the look-up table typically contains a relatively small number of entries along each dimension. With a relatively small table, interpolation is used to convert source coordinates to destination coordinates. The total number of entries in the look-up table can be expressed as $D_dN_s^d$, where d is the dimensionality of the source device color space, D_d is the dimensionality of the destination device color space, and N_s is the number of entries along each dimension of the look-up table. For example, a look-up table that is used to transform color coordinates between two CMYK (*i.e.*, four-dimensional) color spaces can contain 4×17^4, or 334,084 entries.

It should be understood that the look-up table need not have the same number of entries along each dimension. If the look-up table contains N_k entries along each respective dimension, where k ranges from 1 to d, the total number of entries in the look-up table can be expressed as $D_D \prod_{k=1}^{d} N_k$. For example, a look-up table that is used to transform color coordinates between two CMYK color spaces and that is to have fifteen entries along three dimensions and seventeen entries along one dimension contains $4 \times 15 \times 15 \times 15 \times 17$, or 229,500 entries.

In application environments in which it is desirable to further reduce computational and memory requirements, the device link table builder 502 may select only a subset of the total number of entries in each dimension of the look-up table, perform the method loop calculations using that subset, and perform, for example, a spline interpolation to fill in the remaining entries of the look-up table.

The device link table builder 502 provides PCS parameters and source device coordinates to a source device profile interpreter 504. The source device profile interpreter 504 generates source PCS values and provides the source PCS values and the source device coordinates to an error reducer 506. In a specific embodiment, the error reducer 506 is implemented using an error minimization technique. Using the source device coordinates, the error reducer 506 estimates an initial set of destination coordinates that are likely to realize an accurate color match with the source device coordinates. This estimation process may be performed using a relatively simple technique. For example, for estimating destination coordinates in an RGB space corresponding to source coordinates in a CMYK color space, the estimation process may use the following equations:

$$C = 1 - R$$

$$M = 1 - G$$

$$Y = 1 - B$$

Alternatively, the source coordinates can be used to estimate the destination coordinates if the source and destination imaging systems use similar color coordinate spaces.

The error reducer 506 provides the set of estimated destination device coordinates to a destination device profile interpreter 508, which also receives the PCS parameters from the device link table builder 502. The destination device profile interpreter 508 then generates a set of destination PCS values as a function of the estimated destination device coordinates and the PCS parameters and provides the destination PCS values to the error reducer 506. If the error between the destination PCS values and the source PCS values is non-zero, the error reducer 506 uses an error reduction (*e.g.*, an error minimization) technique to reduce the error between the source and destination PCS values. In one embodiment, this is implemented by repeatedly querying the destination device profile interpreter 508 with selected estimates of destination device coordinates. This process can continue until destination device coordinates are found that satisfy a quality threshold, for example, that yield the minimum error. The error reducer 506 returns these destination device coordinates to the device link table builder 502, which enters them in an appropriate location in the look-up table. The device link table builder 502 then enters the next set of table input entries corresponding to a set of source device color coordinates.

For colors within the gamut of the destination device, the error can be reduced using any of a variety of reduction techniques. For example, Powell's method can be used to perform the error reduction or error minimization.

For additional information regarding Powell's method, reference may be made to William H. Press *et al.*, Numerical Recipes in C (1992), pp. 309-315, available from Cambridge University Press, incorporated herein by reference and attached hereto as Appendix A.

Using this approach, the error reducer 506 generally defines an error function having input parameters that can be varied by the error reduction technique. The error reducer 506 then determines the optimal values of the input parameters resulting in a minimal error. To determine the values of destination device coordinates using the minimum PCS error between the source and destination PCS values, the variable input parameters are the destination device coordinates. Accordingly, in this specific implementation, the error reducer 506 defines the error function as:

$$\text{Error}(\mathbf{D}) = \Delta\text{E}(\mathbf{R}_s, \mathbf{R}_d(\mathbf{D}))$$

where **D** is a vector defined by the destination device coordinates, $\mathbf{R}_s$ is a vector defined by the source PCS values and $\mathbf{R}_d(\mathbf{D})$ is a vector function producing destination PCS values as a function of the destination coordinate vector **D**, and ΔE is the Euclidean distance error between $\mathbf{R}_s$ and $\mathbf{R}_d(\mathbf{D})$.

The Euclidean distance error may be expressed using the following equation:

$$\Delta E(\mathbf{R}_1, \mathbf{R}_2) = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$$

The above equation assumes that the PCS is implemented as the CIELAB color space. It should be understood, however, that other color spaces may be used as a PCS. For example, one color space that is particularly suited for use as a PCS is described in the previously-referenced copending U.S. Patent Application, entitled "Characterization of Color Imaging Systems."

Using this same approach, a non-zero optimal error indicates that the source device is out of gamut relative to the destination device at that location in the PCS. In such situations, the error reducer 506 optionally uses the destination device coordinates that result in a minimum ΔE value. Alternatively, the error reducer 506 may use these values as an initial estimate and recalculate the optimal destination device coordinates using a new error function that employs weighting factors, polar coordinates in the chromatic plane of the PCS space, or both.

The error reducer 506 optionally uses a gamut mapping parameter received from the device link table builder 502 to decide how to map coordinates that are out of gamut relative to the source device. For example, the gamut mapping parameter may specify modes in which each technique is used for obtaining destination device coordinates. One mode, for example, may use lightness, chroma, and hue values L^*, C^*, and h^* instead of LAB values L^*, a^*, and b^*, where:

$$C^* = \sqrt{\left(a^{*2} + b^{*2}\right)}\text{ , and}$$

$$h^* = C^* \cdot \mathrm{ARCTAN}\left(b^* / a^*\right) .$$

Another mode uses the above lightness, chroma, and hue values as well as weighting factors:

$$\mathrm{Error}(\mathbf{D}) = \Delta\mathrm{EW}(\mathbf{R_s}, \mathbf{R_d}(\mathbf{D}), \mathbf{W})$$

$$\Delta\mathrm{EW}(\mathbf{R_1}, \mathbf{R_2}, \mathbf{W}) = \sqrt{W_L\left(L_1 - L_2\right)^2 + W_C\left(C_1 - C_2\right)^2 + W_h\left(h_1 - h_2\right)^2}$$

where the PCS vectors $\mathbf{R_s}$, $\mathbf{R_d}(\mathbf{D})$ are converted to lightness, chroma, and hue values either before or after passing them to the error function.

If the weighting factors are one, the above-weighted error reduction function ΔEW() is identical to the standard ΔE() error reduction function. It should be noted, however, that weighting factors of W_L=3, W_C=1, and W_h=1.5 yield particularly accurate visual results. These weighting factors produce an error function that gives priority first to lightness, then to hue, then to chroma. These weighting factors can also be provided to the error reducer 506 as gamut mapping parameters by the device link table builder 502.

Creating the device profile link via error reduction of the forward transformations of the devices realizes a number of advantages. For example, errors in color conversion are limited to those attributable to rounding and interpolation. As a result, the cumulative error from repeated forward and reverse transformations between the source and destination device color spaces is substantially reduced. Additionally, the color transformer can select the gamut mapping technique. The color transformer can rely on the forward transform information and realize consistent gamut mapping between device profiles supplied by different vendors. It should be noted that errors due to interpolation of the device profile link decrease as the number of table entries in each dimension of the look-up table increases toward the maximum number of gray levels. This error also decreases if a one-dimensional tone reproduction table is used to transform the color values. For additional information regarding the use of a one-dimensional tone reproduction table, reference is made to U.S. Patent No. 5,432,906, issued to Gary H. Newman, assigned to Eastman Kodak Company, and incorporated by reference.

Creating the device profile link using error reduction also allows transformation between CMYK device spaces that maps the tone response of the source and destination black (K) channels while maintaining an accurate match with the $L^*a^*b^*$ data. For transformation from an RGB source device to a CMYK destination device, the RGB color coordinates used by the source device lack K channel information. Some conventional color transformation techniques use a process known as gray component removal (GCR) to define a relationship between K values and CMY values in the reverse transformation (*i.e.*, $L^*a^*b^*$ to CMYK). For example, the reverse transformation may be performed with K initially set to zero. The value of K can then be calculated based on the minimum of the C, M, and Y values. The CMY values can then be recalculated using an algebraic calculation or using the forward model to obtain the closest value of $L^*a^*b^*$ input using the new calculated K value. This process involves a reverse transformation from $L^*a^*b^*$ color values to CMYK color values with a fixed definition of GCR.

This process, however, loses the K channel information or the CMY channel information during the translation between CMYK color spaces because the source color values are transformed to a three-dimensional intermediary color space during conversion to destination CMYK values. To preserve the K channel information, the error reducer 506 determines optimal K values in the destination color space that correspond to the K values in the source device color space, *e.g.*, values between 0 and 255. These values can be created, for example, by generating a series of source K values ranging from minimum to maximum, fixing the source and destination CMY values at 0, and finding destination K values with minimum ΔE error relative to each of the source K values. These source and destination K values can be loaded into a lookup table for quick conversion of source K to destination K values. By using error reduction to determine optimal K values in the destination color space, the device link generator 500 preserves K channel information. This results in improved accuracy of the K channel information when converting colors between CMYK devices.

After loading the source and destination K values into a lookup table, when the error reducer 506 receives source $L^*a^*b^*$ and CMYK values, the error reducer 506 initially maps the source K channel to the destination K channel. The error reduction procedure is then used for varying the destination CMY values to obtain the best match for the respective $L^*a^*b^*$ values. If $\Delta E = 0$, control returns to the device link table builder 502, which enters the calculated destination CMYK values into the device link table. If ΔE is greater than zero, then the destination CMY values corresponding to the destination K value in question are out of gamut relative to the target $L^*a^*b^*$ values. This may be, for example, because the source CMY values corresponding to K = 0 result in a color that is out of gamut with the destination device, or because the destination K value in the particular region of destination CMY color space is either too high or too low, *i.e.*, the mixture of K with CMY is such that the resulting color is too dark or too light relative to the targeted $L^*a^*b^*$ value.

To reduce the ΔE error, K can be varied in a controlled way so as to ensure both optimal $L^*a^*b^*$ color and optimal matching of the K source channel behavior. This can be performed, for example, by alternately fixing the current CMY values while performing error reduction on variable K values and fixing the K value while performing error reduction on variable CMY values. When it is determined that neither varying CMY nor varying K improves the ΔE error, it can be assumed that the optimal CMYK values have been determined to satisfy both the color matching and K channel accuracy criteria. Control then optionally returns to the device link table builder 502. While the above discussion assumes that the error reducer 506 performs the mapping between source and destination K values, it should be understood that the device link table builder 502 can perform the mapping.

It should be understood that other approaches can be used to improve the accuracy of the K channel information. For example, the PCS can be implemented as a color space having the same number of dimensions, *e.g.*, four, as CMYK space. Using a PCS having the same dimensionality as the device space prevents the loss of color channel information. In a specific example embodiment, the first three channels of this PCS are the PCS currently used by the system (*e.g.*, LAB, $L^*a^*b^*$, or XYZ). The fourth channel indicates a PCS value indicative of the black channel or relating to the black channel (*e.g.*, L^* or tristimulus value Y). The process can be performed in a manner similar to that performed by the ICC specification as in, for example, ColorSync 2.1 available from Apple Computer.

FIG. 6 illustrates an example color transformation method 600 according to the present invention. At a block 602, selected source device color coordinates are mapped to a PCS. Destination device color coordinates are then estimated as a function of the source device color coordinates, as depicted at a block 604. These estimated destination device color coordinates are then mapped to the PCS at a block 606.

At a block 608, an error between the PCS values corresponding to the source and destination device color coordinates is determined. At a decision block 610, the method determines whether the error satisfies a quality criterion, such as error minimization. In certain applications, the quality criterion can be defined as reduction of the error below a threshold value. If the error does not satisfy the quality criterion, flow proceeds to a block 612, at which the estimated destination device color coordinates are adjusted to reduce the error. This process repeats until the error is reduced.

After the error is reduced, flow proceeds to a block 614, at which the optimal destination device color coordinates thus obtained are entered into a color map. Next, the method determines whether the color map is filled, as depicted at a decision block 616. If the color map contains empty entries, flow proceeds to a block 618. New source device color coordinates are then selected, and then flow returns to the block 602. This process continues until the color map is filled. The color map can then be stored as, for example, a data file for future reference. The user can specify the desired source, destination, and intermediate profiles and the user preferences used to generate the device profile link. Upon recognizing that a color map has already been developed for a particular combination of these profiles, the system can load the data file. Loading the data file instead of reconstructing the color map saves computation time and other resources.

The device profile link can be generated each time the user requests a new combination of device profiles. Alternatively, the user can specify in advance a series of source, intermediate, and destination profiles and allow the system to preprocess these lists of profiles into their respective device profile links and store them. When the user requests that a particular transform be performed on image data using a previously defined combination of source, intermediate, and destination profiles, the system retrieves the associated device profile link. Retrieving the device profile link improves the processing speed.

While the above discussion has assumed that the device profile link describes a conversion between two device profiles, it should be understood that the device profile link can be used to describe a conversion between any number of device profiles. For example, N device profiles can be concatenated using a single device profile link. To concatenate the device profiles, the color conversion is performed using the PCS to convert colors between each device profile to be concatenated. Performing error reduction on the forward transforms between the individual device profiles improves the accuracy of the concatenated device profile link between the first and n^{th} device profiles.

The various embodiments described above are provided by way of illustration only and should not be construed to limit the invention. Those skilled in the art will readily recognize various modifications and changes that may be made to the present invention without strictly following the example embodiments and applications illustrated and described herein, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.

What is claimed is:

1. For use in transforming colors between color imaging systems, a color mapping method comprising:

using forward transformation profiles that characterize the color imaging systems to generate respective sets of device-independent color values for the color imaging systems;

calculating color conversions by recursively reducing differences between the sets of device-independent color values; and

constructing a color map describing a relationship between the color imaging systems using the color conversions.

2. A color mapping method, according to claim 1, further comprising recursively reducing differences between black channel information.

3. A color mapping method, according to claim 1, further comprising using an error function for calculating the color conversions.

4. A color mapping method, according to claim 1, further comprising configuring at least one of the profiles

to account for certain perceptual effects on color appearance.

5. A color mapping method, according to claim 1, wherein the color map comprises at least one of the following: a lookup table, and an equation.

6. A color mapping method, according to claim 1, further comprising:

storing the color map;

detecting respective types of color imaging devices between which a color transformation is to be performed; and

in response to the detected types, selecting a stored color map.

7. For use in transforming colors between source and destination color imaging systems, a color mapping method comprising:

using profiles that characterize the color imaging systems to generate device-independent color values for the source color imaging system, the device-independent color values having a same dimensionality as the source color imaging system;

using the profiles to perform a color conversion for converting the device-independent color values to device-dependent values of the destination color imaging system; and

using the color conversion to define a color map for transforming colors between the color imaging systems.

8. A color mapping method, according to claim 7, wherein the color conversion is performed at least twice.

9. A color mapping method, according to claim 7, further comprising:

using the color conversion to evaluate its accuracy at least once; and

revising the color conversion at least once to improve its accuracy.

10. For use in transforming colors between source and destination color imaging systems, a color mapping method comprising:

(a) using profiles characterizing the color imaging systems to generate device-independent color values for the source color imaging system, the device-independent color values having a same dimensionality as the source color imaging system;

(b) using the profiles to perform a color conversion for converting the device-independent color values to device-dependent values of the destination color imaging system;

(c) using the color conversion to improve the accuracy of the color conversion relative to a quality threshold;

(d) returning to step (c) until the color conversion satisfies the quality threshold; and

(e) using the color conversion to define a color map for transforming colors between the color imaging systems.

11. For use in transforming colors between color imaging systems, a color mapping arrangement comprising:

means for using forward transformation profiles that characterize the color imaging systems to generate respective sets of device-independent color values for the color imaging systems;

means for calculating color conversions by recursively reducing differences between the sets of device-independent color values; and

means for constructing a color map describing a relationship between the color imaging systems using the color conversions.

12. For use in transforming colors between first and second color imaging systems respectively using first and second color coordinate systems, a color mapping method comprising:

(a) generating first device-independent color coordinates as a function of color coordinates in the first color coordinate system;

(b) estimating preliminary color coordinates in the second color coordinate system;

(c) generating second device-independent color coordinates as a function of the preliminary color coordinates;

(d) adjusting the preliminary color coordinates to reduce an error between the first and second device-independent color coordinates;

(e) returning to step (a) until the error satisfies a quality threshold; and

(f) constructing a color map describing a relationship between the first and second color imaging systems as a function of the adjusted color coordinates.

13. A color mapping method, according to claim 12, further comprising using the color coordinates in the first color coordinate system to estimate the preliminary color coordinates.

14. For use in transforming colors between color imaging systems, a color mapping arrangement comprising:

a computer arrangement, programmed to

use forward transformation profiles that characterize the color imaging systems to generate respective sets of device-independent color values for the color imaging systems, calculate color conversions by recursively reducing differences between the sets of device-independent color values, and

construct a color map describing a relationship between the color imaging systems using the color conversions; and

a memory, configured and arranged to store the color map.

15. A color mapping arrangement, according to claim 14, wherein the computer arrangement is further programmed to use an error function for calculating the color conversions.

16. A color mapping arrangement, according to claim 14, wherein the computer arrangement is further programmed to configure at least one of the profiles to account for certain perceptual effects on color appearance.

17. A color mapping arrangement, according to claim 14, wherein the computer arrangement is further programmed to construct at least one of the following: a lookup table, and an equation.

18. A color mapping arrangement, according to claim 14, wherein the computer arrangement is further programmed to

detect respective types of color imaging devices between which a color transformation is to be performed, and

in response to the detected types, select a stored color map.

19. For use in transforming colors between color imaging systems, a data storage medium storing a computer-executable program that, when executed,

uses forward transformation profiles that characterize the color imaging systems to generate respective sets of device-independent color values for the color imaging systems;

calculates color conversions by recursively reducing differences between the sets of device-independent color values, and

constructs a color map describing a relationship between the color imaging systems using the color conversions.

20. A data storage medium, according to claim 19, wherein the computer-executable program recursively reduces differences between black channel information.

21. A data storage medium, according to claim 19, wherein the computer-executable program uses an error function for calculating the color conversions.

22. A data storage medium, according to claim 19, wherein the computer-executable program configures at least one of the profiles to account for certain perceptual effects on color appearance.

23. A data storage medium, according to claim 19, wherein the computer-executable program generates at least one of the following: a lookup table, and an equation.

24. A data storage medium, according to claim 19, wherein the computer-executable program:

stores the color map;

detects respective types of color imaging devices between which a color transformation is to be performed; and in response to the detected types, selects a stored color map.

Abstract

A color mapping method is used in transforming colors between color imaging systems. The method includes using forward transformation profiles that characterize the color imaging systems to generate respective sets of device-independent color values for the color imaging systems. Color conversions are calculated by recursively reducing differences between the respective sets of device-independent color values. Based on these color conversions, a color map is constructed that describes a relationship between the color imaging systems.

"Express Mail" mailing number EM296798633US

Date of Deposit July 3, 1997

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

Nassir Mohamoud
printed name

signature

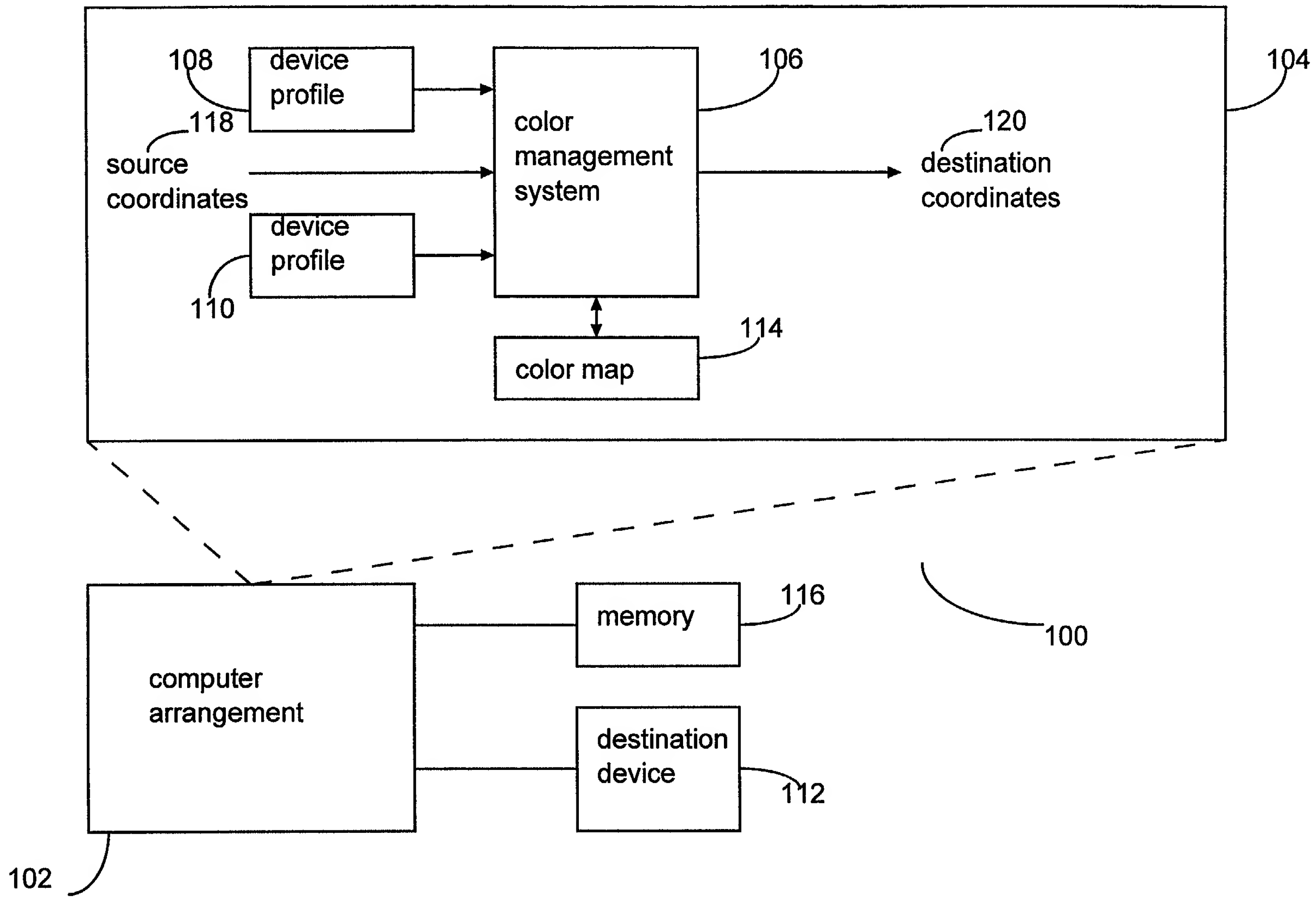

108
device profile
118
source coordinates
device profile
110
color management system
106
color map
114
120
destination coordinates
104
100
computer arrangement
memory
116
destination device
112
102

FIG. 1

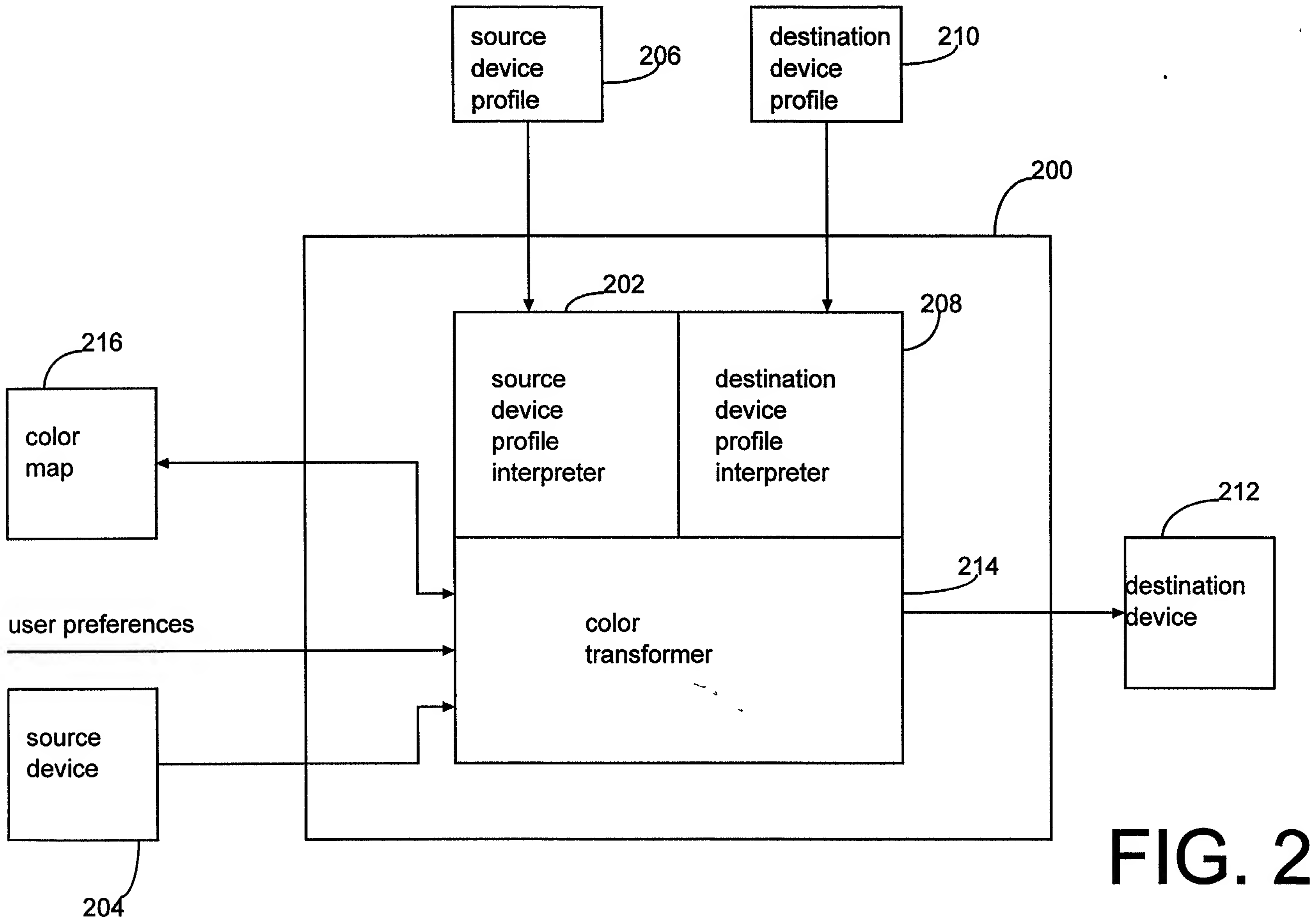

source
device
profile
206
destination
device
profile
210
200
202
208
216
color
map
source
device
profile
interpreter
destination
device
profile
interpreter
212
214
destination
device
user preferences
color
transformer
source
device
204

FIG. 2

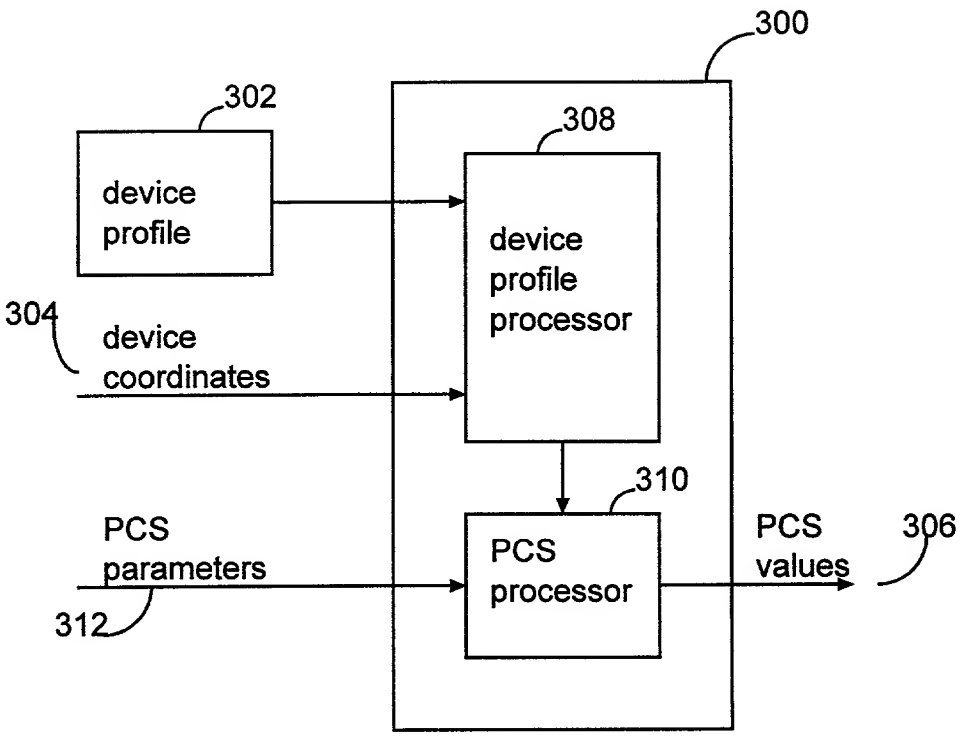

300
302
device profile
308
device profile processor
304
device coordinates
310
PCS parameters
312
PCS processor
PCS values
306

FIG. 3

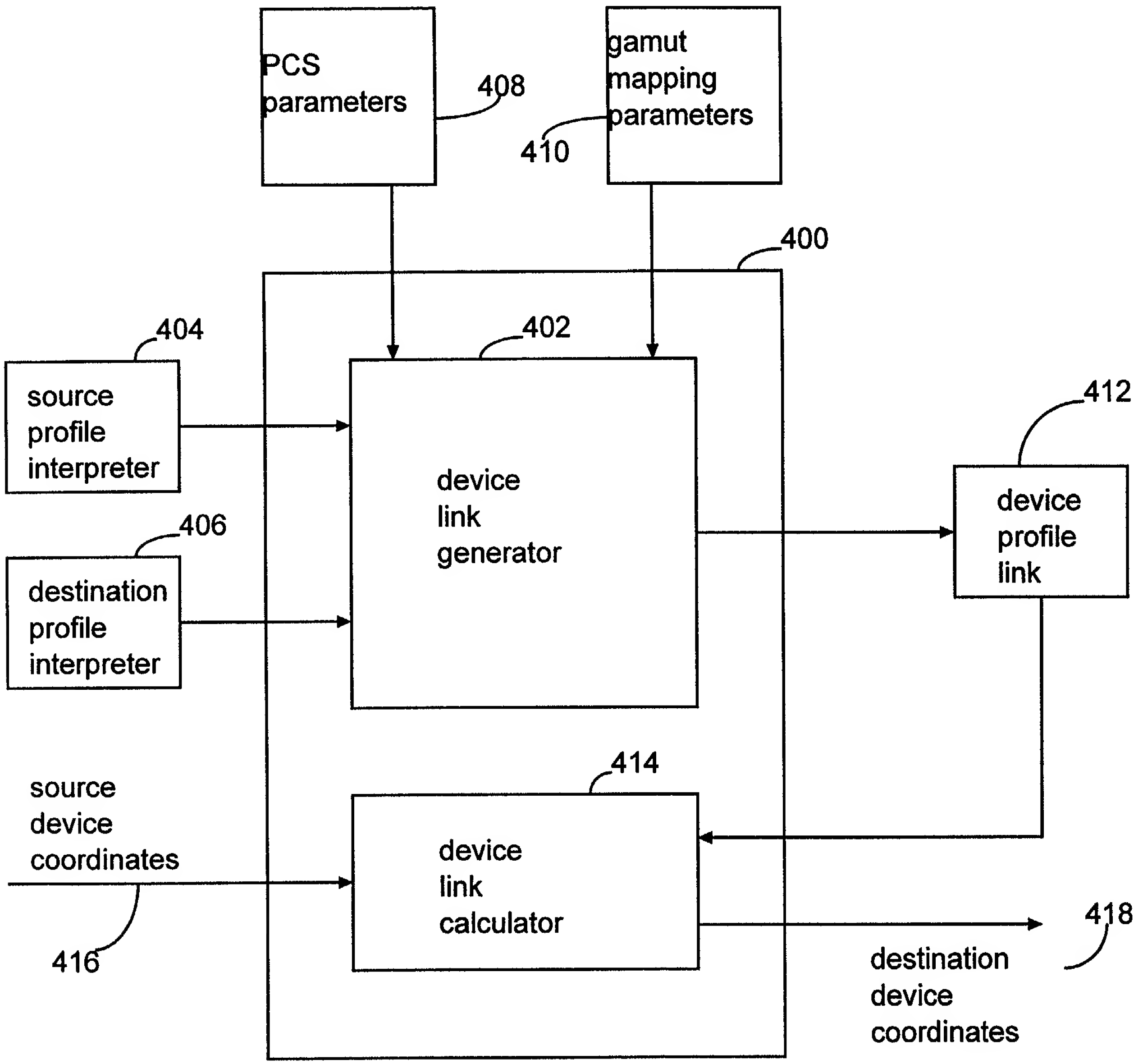

PCS parameters
408
gamut mapping parameters
410
400
404
source profile interpreter
402
device link generator
412
device profile link
406
destination profile interpreter
414
source device coordinates
device link calculator
416
418
destination device coordinates

FIG. 4

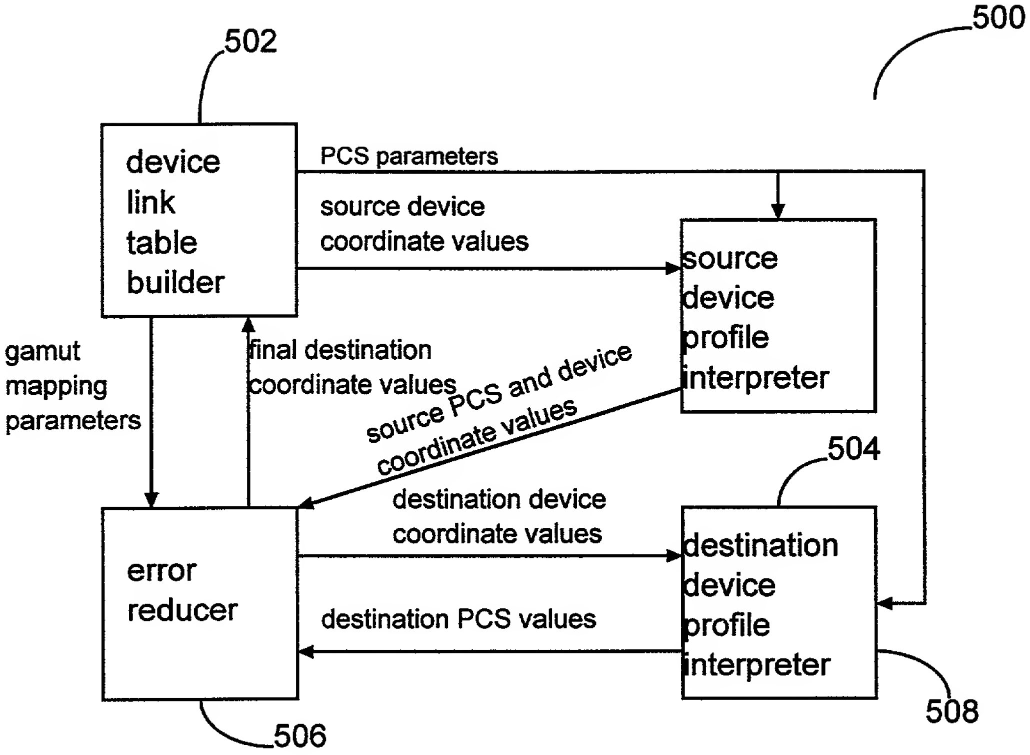

500
502
device link table builder
PCS parameters
source device coordinate values
source device profile interpreter
gamut mapping parameters
final destination coordinate values
source PCS and device coordinate values
504
destination device coordinate values
destination device profile interpreter
error reducer
destination PCS values
506
508

FIG. 5

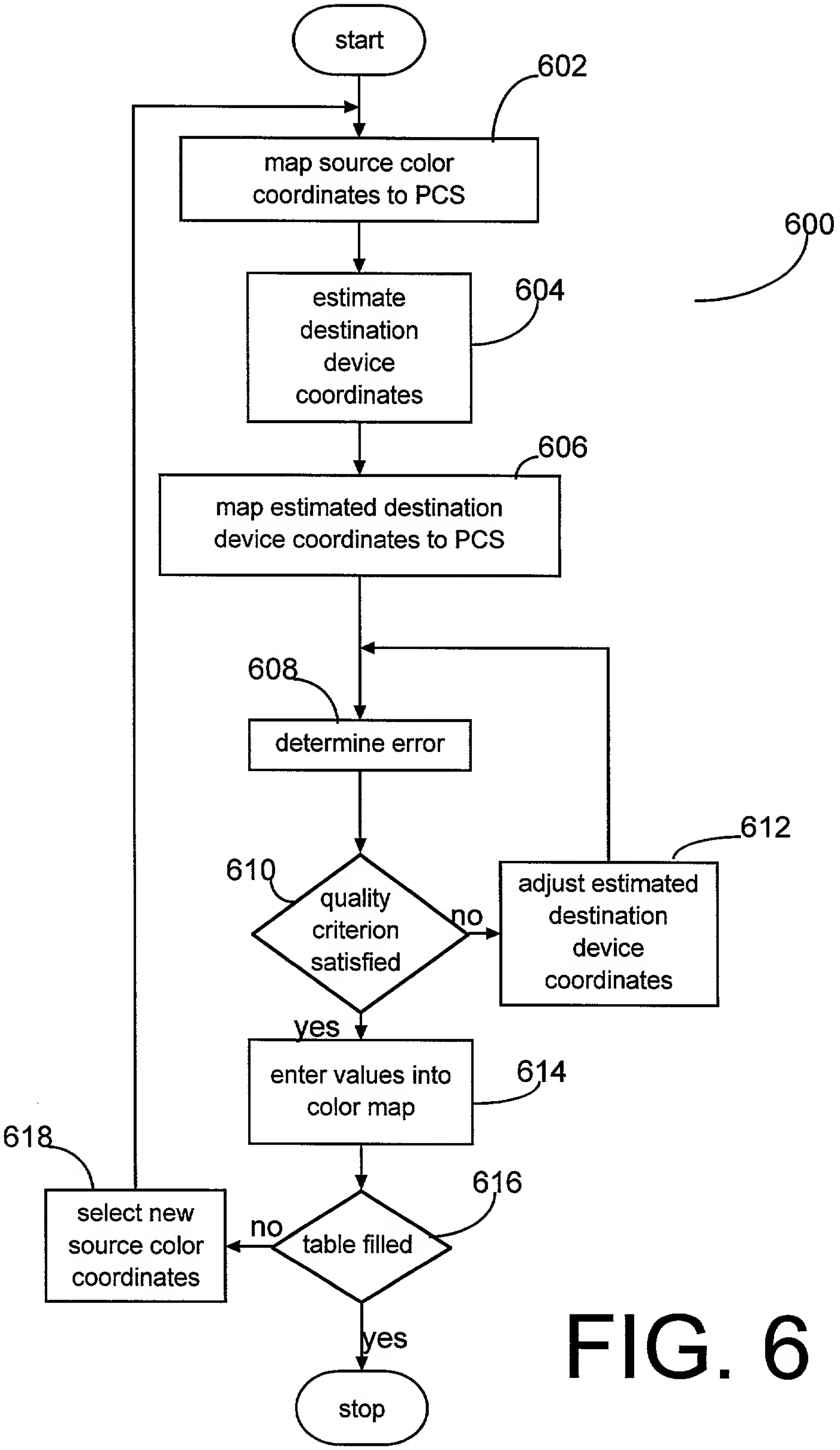

start
602
map source color coordinates to PCS
600
estimate destination device coordinates
604
606
map estimated destination device coordinates to PCS
608
determine error
610
quality criterion satisfied
no
612
adjust estimated destination device coordinates
yes
enter values into color map
614
618
select new source color coordinates
no
table filled
616
yes
stop

FIG. 6

MERCHANT, GOULD, SMITH, EDELL, WELTER & SCHMIDT

United States Patent Application

DECLARATION

As a below named inventor I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; that

I verily believe I am the original, first and sole inventor (if only one name is listed below) or a joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: ARRANGEMENT FOR MAPPING COLORS BETWEEN IMAGING SYSTEMS AND METHOD THEREFOR.

The specification of which:

a. □ is attached hereto
b. ☒ is entitled ARRANGEMENT FOR MAPPING COLORS BETWEEN IMAGING SYSTEMS AND METHOD THEREFOR, having an attorney docket number of 4362.37US02.
c. □ was filed on as application serial no. and was amended on (if applicable) (in the case of a PCT-filed application) described and claimed in international no. filed and as amended on (if any), which I have reviewed and for which I solicit a United States patent.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, § 1.56 (attached hereto).

I hereby claim foreign priority benefits under Title 35, United States Code, § 119/365 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on the basis of which priority is claimed:

a. ☒ no such applications have been filed.
b. □ such applications have been filed as follows:

FOREIGN APPLICATION(S), IF ANY, CLAIMING PRIORITY UNDER 35 USC § 119			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)
ALL FOREIGN APPLICATION(S), IF ANY, FILED BEFORE THE PRIORITY APPLICATION(S)			
COUNTRY	APPLICATION NUMBER	DATE OF FILING (day, month, year)	DATE OF ISSUE (day, month, year)

I hereby claim the benefit under Title 35, United States Code, § 120/365 of any United States and PCT international application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. APPLICATION NUMBER	DATE OF FILING (day, month, year)	STATUS (patented, pending, abandoned)

I hereby claim the benefit under Title 35, United States Code § 119(e) of any United States provisional application(s) listed below:

U.S. PROVISIONAL APPLICATION NUMBER	DATE OF FILING (Day, Month, Year)

Please direct all correspondence in this case to Merchant, Gould, Smith, Edell, Welter & Schmidt at the address indicated below:

Merchant, Gould, Smith, Edell,
Welter & Schmidt
3100 Norwest Center
90 South Seventh Street
Minneapolis, MN 55402-4131

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2	**Full Name** **Of Inventor**	**Family Name** EDGE	**First Given Name** CHRISTOPHER	**Second Given Name** J.
0	**Residence** **& Citizenship**	**City** ST. PAUL	**State or Foreign Country** MINNESOTA	**Country of Citizenship** U.S.A.
1	**Post Office** **Address**	**Post Office Address** 1890 JEFFERSON AVENUE	**City** ST. PAUL	**State & Zip Code/Country** MINNESOTA 55105/U.S.A.
Signature of Inventor 201:			**Date:**	
2	**Full Name** **Of Inventor**	**Family Name** FISCHER	**First Given Name** TIMOTHY	**Second Given Name** A.
0	**Residence** **& Citizenship**	**City** MENDOTA HEIGHTS	**State or Foreign Country** MINNESOTA	**Country of Citizenship** U.S.A.
1	**Post Office** **Address**	**Post Office Address** 2255 FIELDSTONE DRIVE	**City** MENDOTA HEIGHTS	**State & Zip Code/Country** MINNESOTA 55120/U.S.A.
Signature of Inventor 201:			**Date:**	
2	**Full Name** **Of Inventor**	**Family Name** ROZZI	**First Given Name** WILLIAM	**Second Given Name** A.
0	**Residence** **& Citizenship**	**City** WEST LAKELAND TOWNSHIP	**State or Foreign Country** MINNESOTA	**Country of Citizenship** U.S.A.
2	**Post Office** **Address**	**Post Office Address** 730 NORELL AVENUE NORTH	**City** WEST LAKELAND TOWNSHIP	**State & Zip Code/Country** MINNESOTA 55082/U.S.A.
Signature of Inventor 202:			**Date:**	

§ 1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§ 1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.

Appendix A

CITED REFERENCES AND FURTHER READING:

Hammersley, J.M., and Handscomb, D.C. 1964, *Monte Carlo Methods* (London: Methuen).

Shreider, Yu. A. (ed.) 1966, *The Monte Carlo Method* (Oxford: Pergamon).

Sobol', I.M. 1974, *The Monte Carlo Method* (Chicago: University of Chicago Press).

Kalos, M.H., and Whitlock, P.A. 1986, *Monte Carlo Methods* (New York: Wiley).

7.7 Quasi- (that is, Sub-) Random Sequences

We have just seen that choosing N points uniformly randomly in an n-dimensional space leads to an error term in Monte Carlo integration that decreases as $1/\sqrt{N}$. In essence, each new point sampled adds linearly to an accumulated sum that will become the function average, and also linearly to an accumulated sum of squares that will become the variance (equation 7.6.2). The estimated error comes from the square root of this variance, hence the power $N^{-1/2}$.

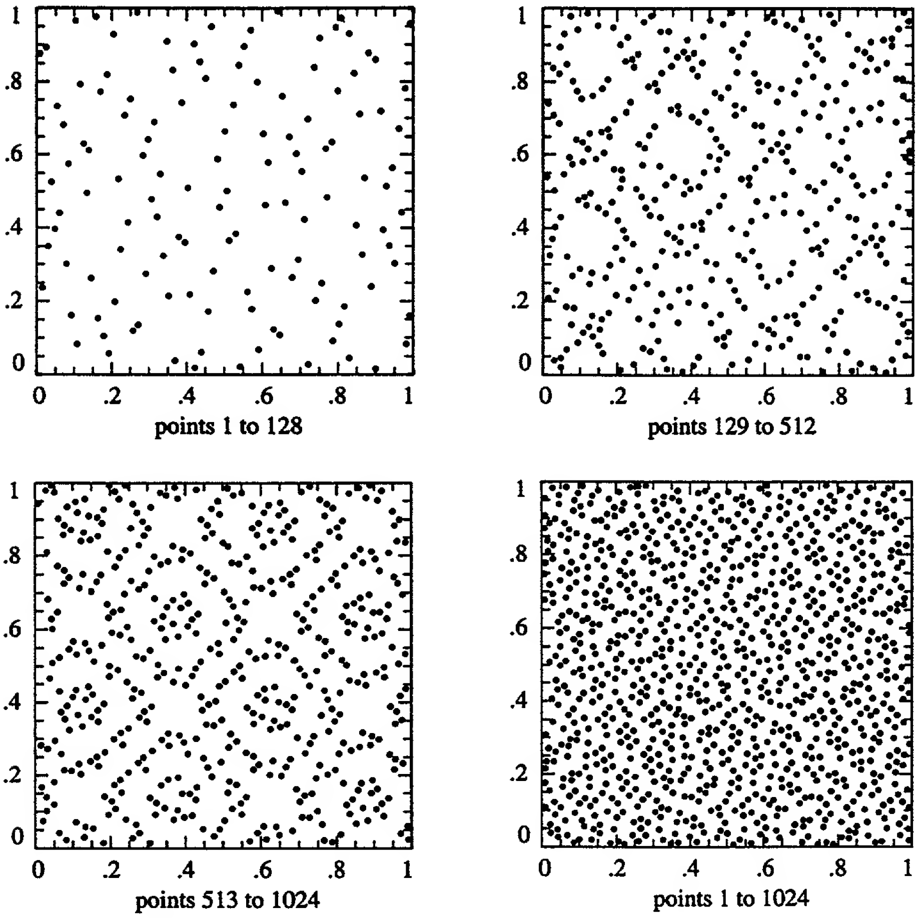
Just because this square root convergence is familiar does not, however, mean that it is inevitable. A simple counterexample is to choose sample points that lie on a Cartesian grid, and to sample each grid point exactly once (in whatever order). The Monte Carlo method thus becomes a deterministic quadrature scheme — albeit a simple one — whose fractional error decreases at least as fast as N^{-1} (even faster if the function goes to zero smoothly at the boundaries of the sampled region, or is periodic in the region).

The trouble with a grid is that one has to decide *in advance* how fine it should be. One is then committed to completing all of its sample points. With a grid, it is not convenient to "sample *until*" some convergence or termination criterion is met. One might ask if there is not some intermediate scheme, some way to pick sample points "at random," yet spread out in some self-avoiding way, avoiding the chance clustering that occurs with uniformly random points.

A similar question arises for tasks other than Monte Carlo integration. We might want to search an n-dimensional space for a point where some (locally computable) condition holds. Of course, for the task to be computationally meaningful, there had better be continuity, so that the desired condition will hold in some finite n-dimensional neighborhood. We may not know *a priori* how large that neighborhood is, however. We want to "sample *until*" the desired point is found, moving smoothly to finer scales with increasing samples. Is there any way to do this that is better than uncorrelated, random samples?

The answer to the above question is "yes." Sequences of n-tuples that fill n-space more uniformly than uncorrelated random points are called *quasi-random sequences*. That term is somewhat of a misnomer, since there is nothing "random" about quasi-random sequences: They are cleverly crafted to be, in fact, *sub*-random. The sample points in a quasi-random sequence are, in a precise sense, "maximally avoiding" of each other.

A conceptually simple example is *Halton's sequence* [1]. In one dimension, the jth number H_j in the sequence is obtained by the following steps: (i) Write j as a number in base b, where b is some prime. (For example $j = 17$ in base $b = 3$ is 122.) (ii) Reverse the digits and put a radix point (i.e., a decimal point base b) in



points 1 to 128

points 129 to 512

points 513 to 1024

points 1 to 1024

Figure 7.7.1. First 1024 points of a two-dimensional Sobol' sequence. The sequence is generated number-theoretically, rather than randomly, so successive points at any stage "know" how to fill in the gaps in the previously generated distribution.

front of the sequence. (In the example, we get 0.221 base 3.) The result is H_j. To get a sequence of n-tuples in n-space, you make each component a Halton sequence with a different prime base b. Typically, the first n primes are used.

It is not hard to see how Halton's sequence works: Every time the number of digits in j increases by one place, j's digit-reversed fraction becomes a factor of b finer-meshed. Thus the process is one of filling in all the points on a sequence of finer and finer Cartesian grids — and in a kind of maximally spread-out order on each grid (since, e.g., the most rapidly changing digit in j controls the *most* significant digit of the fraction).

Other ways of generating quasi-random sequences have been suggested by Faure, Sobol', Niederreiter, and others. Bratley and Fox [2] provide a good review and references, and discuss a particularly efficient variant of the Sobol' [3] sequence suggested by Antonov and Saleev [4]. It is this Antonov-Saleev variant whose implementation we now discuss.

Degree	Primitive Polynomials Modulo 2*
1	0 (i.e., $x + 1$)
2	1 (i.e., $x^2 + x + 1$)
3	1, 2 (i.e., $x^3 + x + 1$ and $x^3 + x^2 + 1$)
4	1, 4 (i.e., $x^4 + x + 1$ and $x^4 + x^3 + 1$)
5	2, 4, 7, 11, 13, 14
6	1, 13, 16, 19, 22, 25
7	1, 4, 7, 8, 14, 19, 21, 28, 31, 32, 37, 41, 42, 50, 55, 56, 59, 62
8	14, 21, 22, 38, 47, 49, 50, 52, 56, 67, 70, 84, 97, 103, 115, 122
9	8, 13, 16, 22, 25, 44, 47, 52, 55, 59, 62, 67, 74, 81, 82, 87, 91, 94, 103, 104, 109, 122, 124, 137, 138, 143, 145, 152, 157, 167, 173, 176, 181, 182, 185, 191, 194, 199, 218, 220, 227, 229, 230, 234, 236, 241, 244, 253
10	4, 13, 19, 22, 50, 55, 64, 69, 98, 107, 115, 121, 127, 134, 140, 145, 152, 158, 161, 171, 181, 194, 199, 203, 208, 227, 242, 251, 253, 265, 266, 274, 283, 289, 295, 301, 316, 319, 324, 346, 352, 361, 367, 382, 395, 398, 400, 412, 419, 422, 426, 428, 433, 446, 454, 457, 472, 493, 505, 508

*Expressed as a decimal integer representing the interior bits (that is, omitting the high-order bit and the unit bit).

The Sobol' sequence generates numbers between zero and one directly as binary fractions of length w bits, from a set of w special binary fractions, V_i, $i = 1, 2, \dots, w$, called *direction numbers*. In Sobol's original method, the jth number X_j is generated by XORing (bitwise exclusive or) together the set of V_i's satisfying the criterion on i, "the ith bit of j is nonzero." As j increments, in other words, different ones of the V_i's flash in and out of X_j on different time scales. V_1 alternates between being present and absent most quickly, while V_k goes from present to absent (or vice versa) only every $2^{k\cdot\ 1}$ steps.

Antonov and Saleev's contribution was to show that instead of using the bits of the integer j to select direction numbers, one could just as well use the bits of the *Gray code* of j, $G(j)$. (For a quick review of Gray codes, look at §20.2.)

Now $G(j)$ and $G(j + 1)$ differ in exactly one bit position, namely in the position of the rightmost zero bit in the binary representation of j (adding a leading zero to j if necessary). A consequence is that the $j + 1$st Sobol'-Antonov-Saleev number can be obtained from the jth by XORing it with *a single* V_i, namely with i the position of the rightmost zero bit in j. This makes the calculation of the sequence very efficient, as we shall see.

Figure 7.7.1 plots the first 1024 points generated by a two-dimensional Sobol' sequence. One sees that successive points do "know" about the gaps left previously, and keep filling them in, hierarchically.

We have deferred to this point a discussion of how the direction numbers V_i are generated. Some nontrivial mathematics is involved in that, so we will content ourself with a cookbook summary only: Each different Sobol' sequence (or component of an n-dimensional sequence) is based on a different primitive polynomial over the integers modulo 2, that is, a polynomial whose coefficients are either 0 or 1, and which cannot be factored (using modulo 2 integer arithmetic) into polynomials of lower order. (Primitive polynomials modulo 2 were used in §7.4, and are further discussed in §20.3.) Suppose P is such a polynomial, of degree q,

$$P = x^q + a_1 x^{q\cdot\ 1} + a_2 x^{q\cdot\ 2} + \cdots + a_{q.\ 1} + 1 \qquad (7.7.1)$$

Initializing Values Used in sobseq					
Degree	**Polynomial**	**Starting Values**			
1	0	1	(3)	(5)	(15) ...
2	1	1	1	(7)	(11) ...
3	1	1	3	7	(5) ...
3	2	1	3	3	(15) ...
4	1	1	1	3	13 ...
4	4	1	1	5	9 ...

Parenthesized values are not freely specifiable, but are forced by the required recurrence for this degree.

Define a sequence of integers M_i by the q-term recurrence relation,

$$M_i = 2a_1M_{i\ 1} \oplus 2^2a_2M_{i\ 2} \oplus \cdots \oplus 2^{q\ 1}M_{i\ q+1}a_{q\ 1} \oplus (2^qM_{i\ q} \oplus M_{i\ q}) \qquad (7.7.2)$$

Here bitwise XOR is denoted by $\oplus$. The starting values for this recurrence are that $M_1, \ldots, M_q$ can be arbitrary odd integers less than $2, \ldots, 2^q$, respectively. Then, the direction numbers V_i are given by

$$V_i = M_i/2^i \qquad i = 1, \ldots, w \qquad (7.7.3)$$

The accompanying table lists all primitive polynomials modulo 2 with degree $q \leq 10$. Since the coefficients are either 0 or 1, and since the coefficients of x^q and of 1 are predictably 1, it is convenient to denote a polynomial by its middle coefficients taken as the bits of a binary number (higher powers of x being more significant bits). The table uses this convention.

Turn now to the implementation of the Sobol' sequence. Successive calls to the function sobseq (after a preliminary initializing call) return successive points in an n-dimensional Sobol' sequence based on the first n primitive polynomials in the table. As given, the routine is initialized for maximum n of 6 dimensions, and for a word length w of 30 bits. These parameters can be altered by changing MAXBIT ($\equiv w$) and MAXDIM, and by adding more initializing data to the arrays ip (the primitive polynomials from the table), mdeg (their degrees), and iv (the starting values for the recurrence, equation 7.7.2). A second table, above, elucidates the initializing data in the routine.

```
#include "nrutil.h"
#define MAXBIT 30
#define MAXDIM 6

void sobseq(int *n, float x[])
When n is negative, internally initializes a set of MAXBIT direction numbers for each of MAXDIM
different Sobol' sequences. When n is positive (but ≤MAXDIM), returns as the vector x[1..n]
the next values from n of these sequences. (n must not be changed between initializations.)
{
    int j,k,l;
    unsigned long i,im,ipp;
    static float fac;
    static unsigned long in,ix[MAXDIM+1],*iu[MAXBIT+1];
    static unsigned long mdeg[MAXDIM+1]={0,1,2,3,3,4,4};
    static unsigned long ip[MAXDIM+1]={0,0,1,1,2,1,4};
    static unsigned long iv[MAXDIM*MAXBIT+1]={
        0,1,1,1,1,1,1,3,1,3,3,1,1,5,7,7,3,3,5,15,11,5,15,13,9};

    if (*n < 0) {                               Initialize, don't return a vector.
        for (k=1;k<=MAXDIM;k++) ix[k]=0;
        in=0;
        if (iv[1] != 1) return;
        fac=1.0/(1L << MAXBIT);
        for (j=1,k=0;j<=MAXBIT;j++,k+=MAXDIM) iu[j] = &iv[k];
        To allow both 1D and 2D addressing.
        for (k=1;k<=MAXDIM;k++) {
            for (j=1;j<=mdeg[k];j++) iu[j][k] <<= (MAXBIT-j);
            Stored values only require normalization.
            for (j=mdeg[k]+1;j<=MAXBIT;j++) {       Use the recurrence to get other val-
                ipp=ip[k];                              ues.
                i=iu[j-mdeg[k]][k];
                i ^= (i >> mdeg[k]);
                for (l=mdeg[k]-1;l>=1;l--) {
                    if (ipp & 1) i ^= iu[j-l][k];
                    ipp >>= 1;
                }
                iu[j][k]=i;
            }
        }
    } else {                                        Calculate the next vector in the se-
        im=in++;                                        quence.
        for (j=1;j<=MAXBIT;j++) {                   Find the rightmost zero bit.
            if (!(im & 1)) break;
            im >>= 1;
        }
        if (j > MAXBIT) nrerror("MAXBIT too small in sobseq");
        im=(j-1)*MAXDIM;
        for (k=1;k<=IMIN(*n,MAXDIM);k++) {          XOR the appropriate direction num-
            ix[k] ^= iv[im+k];                          ber into each component of the
            x[k]=ix[k]*fac;                             vector and convert to a floating
        }                                               number.
    }
}
```

How good is a Sobol' sequence, anyway? For Monte Carlo integration of a smooth function in n dimensions, the answer is that the fractional error will decrease with N, the number of samples, as $(\ln N)^n/N$, i.e., almost as fast as $1/N$. As an example, let us integrate a function that is nonzero inside a torus (doughnut) in three-dimensional space. If the major radius of the torus is R_0, the minor radial coordinate r is defined by

$$r = \left([(x^2+y^2)^{1/2} - R_0]^2 + z^2 \right)^{1/2} \tag{7.7.4}$$

Let us try the function

$$f(x,y,z) = \begin{cases} 1 + \cos\left(\dfrac{\pi r^2}{a^2}\right) & r < r_0 \\ 0 & r \geq r_0 \end{cases} \tag{7.7.5}$$

which can be integrated analytically in cylindrical coordinates, giving

$$\int\int\int dx\,dy\,dz\, f(x,y,z) = 2\pi^2 a^2 R_0 \tag{7.7.6}$$

With parameters $R_0 = 0.6$, $r_0 = 0.3$, we did 100 successive Monte Carlo integrations of equation (7.7.4), sampling uniformly in the region $-1 < x, y, z < 1$, for the two cases of uncorrelated random points and the Sobol' sequence generated by the routine sobseq. Figure 7.7.2 shows the results, plotting the r.m.s. average error of the 100 integrations as a function of the number of points sampled. (For any *single* integration, the error of course wanders from positive to negative, or vice versa, so a logarithmic plot of fractional error is not very informative.) The thin, dashed curve corresponds to uncorrelated random points and shows the familiar $N^{-1/2}$ asymptotics. The thin, solid gray curve shows the result for the Sobol'

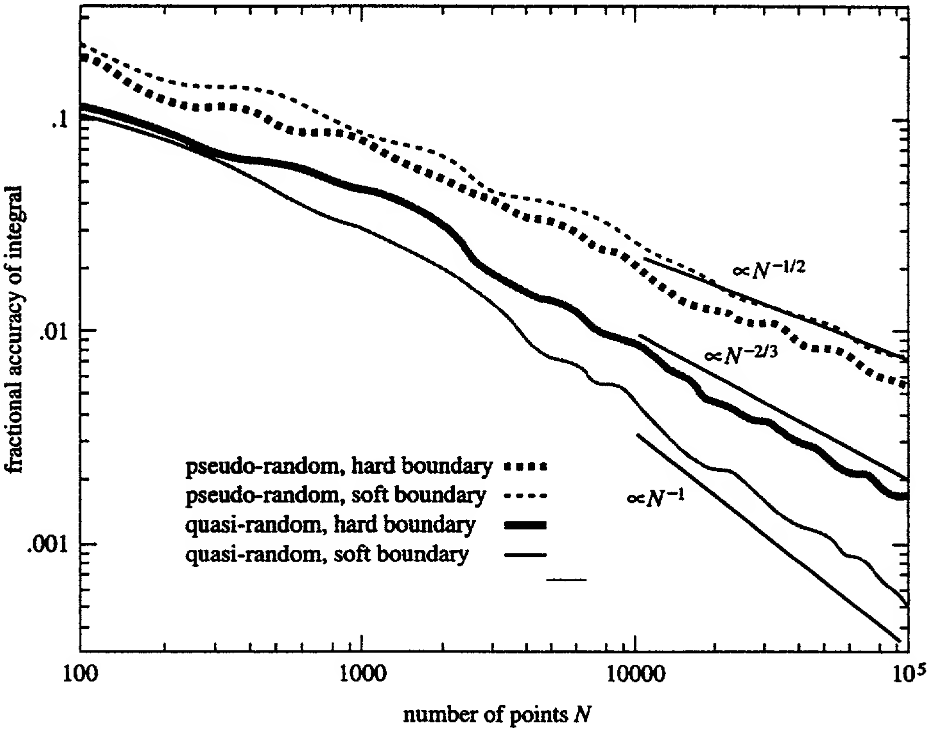


Figure 7.7.2. Fractional accuracy of Monte Carlo integrations as a function of number of points sampled, for two different integrands and two different methods of choosing random points. The quasi-random Sobol' sequence converges much more rapidly than a conventional pseudo-random sequence. Quasi-random sampling does better when the integrand is smooth ("soft boundary") than when it has step discontinuities ("hard boundary"). The curves shown are the r.m.s. average of 100 trials.

sequence. The logarithmic term in the expected $(\ln N)^3/N$ is readily apparent as curvature in the curve, but the asymptotic N^{-1} is unmistakable.

To understand the importance of Figure 7.7.2, suppose that a Monte Carlo integration of f with 1% accuracy is desired. The Sobol' sequence achieves this accuracy in a few thousand samples, while pseudorandom sampling requires nearly 100,000 samples. The ratio would be even greater for higher desired accuracies.

A different, not quite so favorable, case occurs when the function being integrated has hard (discontinuous) boundaries inside the sampling region, for example the function that is one inside the torus, zero outside,

$$f(x,y,z) = \begin{cases} 1 & r < r_0 \\ 0 & r \geq r_0 \end{cases} \qquad (7.7.7)$$

where r is defined in equation (7.7.4). Not by coincidence, this function has the same analytic integral as the function of equation (7.7.5), namely $2\pi^2 a^2 R_0$.

The carefully hierarchical Sobol' sequence is based on a set of Cartesian grids, but the boundary of the torus has no particular relation to those grids. The result is that it is essentially random whether sampled points in a thin layer at the surface of the torus, containing on the order of $N^{2/3}$ points, come out to be inside, or outside, the torus. The square root law, applied to this thin layer, gives $N^{1/3}$ fluctuations in the sum, or $N^{-2/3}$ fractional error in the Monte Carlo integral. One sees this behavior verified in Figure 7.7.2 by the thicker gray curve. The thicker dashed curve in Figure 7.7.2 is the result of integrating the function of equation (7.7.7) using independent random points. While the advantage of the Sobol' sequence is not quite so dramatic as in the case of a smooth function, it can nonetheless be a significant factor (~5) even at modest accuracies like 1%, and greater at higher accuracies.

Note that we have not provided the routine `sobseq` with a means of starting the sequence at a point other than the beginning, but this feature would be easy to add. Once the initialization of the direction numbers `iv` has been done, the jth point can be obtained directly by XORing together those direction numbers corresponding to nonzero bits in the Gray code of j, as described above.

The Latin Hypercube

We might here give passing mention the unrelated technique of *Latin square* or *Latin hypercube* sampling, which is useful when you must sample an N-dimensional space *exceedingly* sparsely, at M points. For example, you may want to test the crashworthiness of cars as a simultaneous function of 4 different design parameters, but with a budget of only three expendable cars. (The issue is not whether this is a good plan — it isn't — but rather how to make the best of the situation!)

The idea is to partition each design parameter (dimension) into M segments, so that the whole space is partitioned into M^N cells. (You can choose the segments in each dimension to be equal or unequal, according to taste.) With 4 parameters and 3 cars, for example, you end up with $3 \times 3 \times 3 \times 3 = 81$ cells.

Next, choose M cells to contain the sample points by the following algorithm: Randomly choose one of the M^N cells for the first point. Now eliminate all cells that agree with this point on *any* of its parameters (that is, cross out all cells in the same row, column, etc.), leaving $(M-1)^N$ candidates. Randomly choose one of these, eliminate new rows and columns, and continue the process until there is only one cell left, which then contains the final sample point.

The result of this construction is that *each* design parameter will have been tested in *every one* of its subranges. If the response of the system under test is dominated by *one* of the design parameters, that parameter will be found with this sampling technique. On the other hand, if there is an important interaction among different design parameters, then the Latin hypercube gives no particular advantage. Use with care.

CITED REFERENCES AND FURTHER READING:

Halton, J.H. 1960, *Numerische Mathematik*, vol. 2, pp. 84–90. [1]

Bratley P., and Fox, B.L. 1988, *ACM Transactions on Mathematical Software*, vol. 14, pp. 88–100. [2]

Lambert, J.P. 1988, in *Numerical Mathematics – Singapore 1988*, ISNM vol. 86, R.P. Agarwal, Y.M. Chow, and S.J. Wilson, eds. (Basel: Birkhaüser), pp. 273–284.

Niederreiter, H. 1988, in *Numerical Integration III*, ISNM vol. 85, H. Brass and G. Hämmerlin, eds. (Basel: Birkhaüser), pp. 157–171.

Sobol', I.M. 1967, *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112. [3]

Antonov, I.A., and Saleev, V.M 1979, *USSR Computational Mathematics and Mathematical Physics*, vol. 19, no. 1, pp. 252–256. [4]

Dunn, O.J., and Clark, V.A. 1974, *Applied Statistics: Analysis of Variance and Regression* (New York, Wiley) [discusses Latin Square].

7.8 Adaptive and Recursive Monte Carlo Methods

This section discusses more advanced techniques of Monte Carlo integration. As examples of the use of these techniques, we include two rather different, fairly sophisticated, multidimensional Monte Carlo codes: **vegas** [1,2], and **miser** [3]. The techniques that we discuss all fall under the general rubric of *reduction of variance* (§7.6), but are otherwise quite distinct.

Importance Sampling

The use of *importance sampling* was already implicit in equations (7.6.6) and (7.6.7). We now return to it in a slightly more formal way. Suppose that an integrand f can be written as the product of a function h that is almost constant times another, positive, function g. Then its integral over a multidimensional volume V is

$$\int f\,dV = \int (f/g)\,gdV = \int h\,gdV \tag{7.8.1}$$

In equation (7.6.7) we interpreted equation (7.8.1) as suggesting a change of variable to G, the indefinite integral of g. That made gdV a perfect differential. We then proceeded to use the basic theorem of Monte Carlo integration, equation (7.6.1). A more general interpretation of equation (7.8.1) is that we can integrate f by instead sampling h — not, however, with uniform probability density dV, but rather with nonuniform density gdV. In this second interpretation, the first interpretation follows as the special case, where the *means* of generating the nonuniform sampling of gdV is via the transformation method, using the indefinite integral G (see §7.2).

More directly, one can go back and generalize the basic theorem (7.6.1) to the case of nonuniform sampling: Suppose that points x_i are chosen within the volume V with a probability density p satisfying

$$\int p\,dV = 1 \tag{7.8.2}$$

The generalized fundamental theorem is that the integral of any function f is estimated, using N sample points $x_i, \dots, x_N$, by

$$I \equiv \int f\,dV = \int \frac{f}{p}\,pdV \approx \left\langle \frac{f}{p} \right\rangle \pm \sqrt{\frac{\langle f^2/p^2\rangle - \langle f/p\rangle^2}{N}} \tag{7.8.3}$$

where angle brackets denote arithmetic means over the N points, exactly as in equation (7.6.2). As in equation (7.6.1), the "plus-or-minus" term is a one standard deviation error estimate. Notice that equation (7.6.1) is in fact the special case of equation (7.8.3), with $p = \text{constant} = 1/V$.

What is the best choice for the sampling density p? Intuitively, we have already seen that the idea is to make $h = f/p$ as close to constant as possible. We can be more rigorous by focusing on the numerator inside the square root in equation (7.8.3), which is the variance per sample point. Both angle brackets are themselves Monte Carlo estimators of integrals, so we can write

$$S \equiv \left\langle \frac{f^2}{p^2} \right\rangle - \left\langle \frac{f}{p} \right\rangle^2 \approx \int \frac{f^2}{p^2}\,pdV - \left[\int \frac{f}{p}\,pdV\right]^2 = \int \frac{f^2}{p}\,dV - \left[\int f\,dV\right]^2 \tag{7.8.4}$$

We now find the optimal p subject to the constraint equation (7.8.2) by the functional variation

$$0 = \frac{\delta}{\delta p} \quad \int \frac{f^2}{p}\,dV - \left[\int f\,dV\right]^2 + \lambda \int p\,dV \Big) \tag{7.8.5}$$